

Enterprise Service Bus (ESB) Product Evaluation Comparisons

Prepared by Robert Woolley
October 18, 2006



UTAH DEPARTMENT OF TECHNOLOGY SERVICES
Office of the Chief Technology Officer
1 State Office Building, 6th Floor
Salt Lake City, Utah 84114

October 18, 2006
Copyright © 2006 State of Utah
All Rights Reserve

CONTENTS

Executive Summary ...	5
Introduction ...	6
Problem ...	9
Premise ...	9
ESB Characteristics ...	10
Key ESB Benefits ...	11
SOA/ESB Governance ...	13
Reality ...	14
Alternative Solutions ...	15
Evaluation Scope and Architectural Premise ...	17
Product Comparison Information ...	20
Vendor Profiles ...	22
Architecture Premise Alignment ...	25
Department of Technology Services ESB Comparisons ...	27
Financial Analysis and Cost Recovery ...	32
Conclusions ...	34
Appendix 1: Scenario/Use Case Based Product Comparisons ...	35
Criteria and Capabilities ...	36
Integration Scenario Evaluation ...	37
Design, Development, and Deployment Evaluation ...	40
Management and Monitoring Evaluation ...	41
Architecture Evaluation ...	43
Product Viability ...	46
Company Viability ...	48
Definitions ...	50
References ...	56

ENTERPRISE SERVICE BUS: PRODUCT EVALUATION COMPARISONS

EXECUTIVE SUMMARY

The Enterprise Service Bus (ESB) is the foundational component for an effective Service Oriented Architecture (SOA). An ESB provides secure interoperability and message transport services between applications using a variety of Web services and related technologies. The result is a loosely coupled, interoperable set of business services that can be developed once and easily shared within the State enterprise.

Traditional development methodologies have focused on the creation of application asset silos within agencies, with significant redundancies of effort in creating similar services in areas such as security and data access. The national development trend is to move away from silos toward the creation and deployment of service-oriented application assets. Over 90% of companies surveyed have projects moving in this direction, and a number of other state and municipal governments have active ESB projects underway.

From a financial and management perspective, the State needs to find ways to reduce the complexity of application development and integration through more efficient consumption of resources. Agencies require that IT become more responsive to changing business needs at a lower cost and in a timelier manner. Because of this, ESB implementation represents a significant business value to the State.

ESB application infrastructure is available from a wide variety of vendors, including both commercial and open source alternatives. Deployment of ESB services can be accomplished within existing agency application frameworks and infrastructures and as a central service for all interested agencies. However, there are specific requirements for governance and agency collaboration that are required to make an ESB and related services usable and effective. Comparisons of vendor offerings indicate that there are multiple alternatives for ESB and SOA infrastructure that could meet the needs of the State and move the State's current development environment toward more modern methodologies with significant benefits to agencies.

This study recommends a formal proof of concept and evaluation of at least three vendor products, including open source and commercial alternatives, using a scenario based methodology. Selection of vendors can be based upon comparison information presented within this report.

INTRODUCTION

An Enterprise Service Bus (ESB) is characterized by many analysts as the foundation of a successful Service Oriented Architecture (SOA). Defined as middleware that provides secure interoperability and message transport services between applications within a SOA, an ESB uses XML, Web services interfaces, messaging adopters, and rules-based routing to create a loosely coupled, interoperable set of business services that can be easily shared within and between enterprises.^{1, 2} Such a service oriented architecture implementation is presented in Figure 1, illustrating where the State is today, with many application asset silos migrating to service oriented application assets.

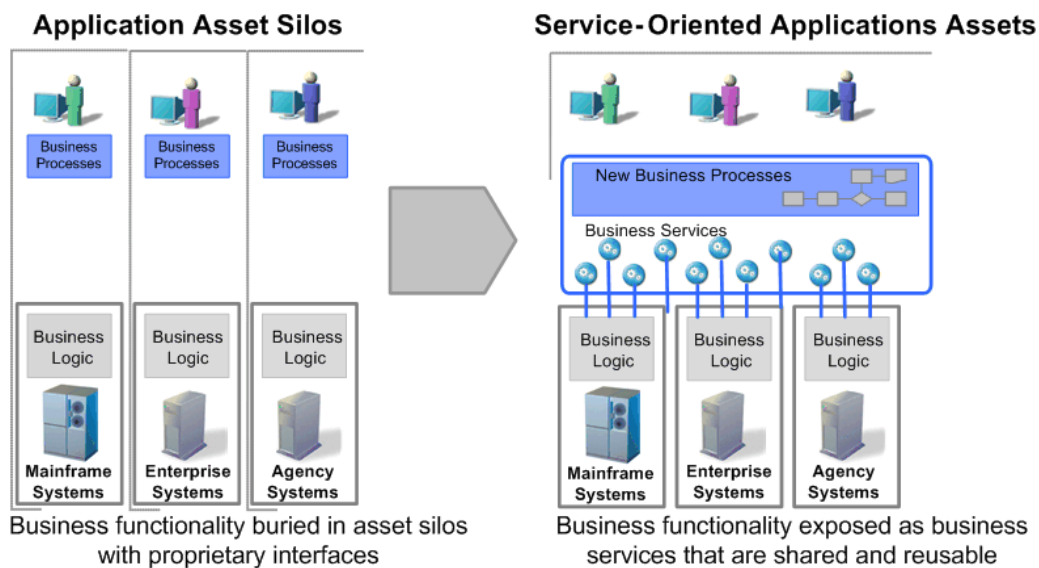


Figure 1. SOA Vision: Evolving to a Services Orientation

The architecture and some of the possible service components of an ESB, from a logical perspective, are illustrated in Figure 2.

¹ *Enterprise Service Bus and SOA Middleware*, Boston: Aberdeen Group, June 2006, p. 1.

² *BEA Overview*, PowerPoint Presentation prepared by Tony Galindo, October 10, 2006.

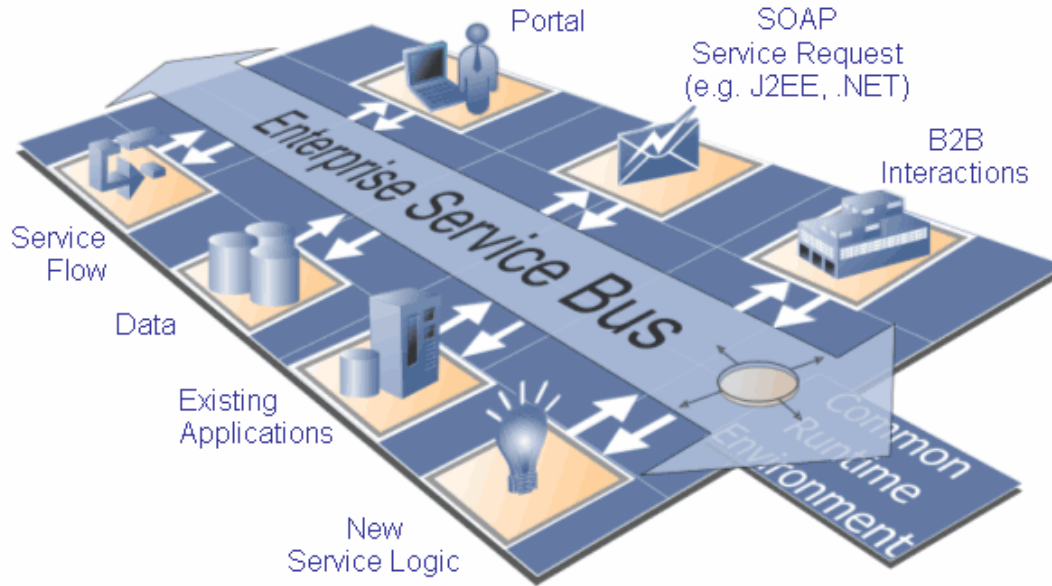


Figure 2. Basic ESB Architecture: Loose Coupling to Data and Shared Services

In this service model an ESB can be deployed to support both autonomous and federated hub and spoke integration requirements. For the State of Utah this represents an important level of architectural and deployment advantage.

Aberdeen split ESB and SOA users into three categories:³

- **SOA Lite:** This category focuses primarily on users that work with .NET and Java with open source SOA software using Eclipse integrated development environments, UDDI registry, SOAP, Ajax, and WS-* standards.
- **Enterprise SOA:** This category focuses on large enterprises that have used SOA for more than a year, with high standards for uptime and availability, and complex integration and legacy computing environments.
- **SOA ERP:** This category focuses on users of SAP and Oracle applications that use those platforms' ESB capabilities as their primary ESB environment.

From a complexity perspective, the State of Utah is clearly an Enterprise SOA environment, but, from an implementation and adoption perspective, from within the agencies, the State is most accurately categorized as SOA Lite. Current development and use of SOA functionality is much more pronounced at the edge at customer sites than as any kind of large, central, single ESB consumed by everyone.

³ *Enterprise Service Bus and SOA Middleware*, Boston: Aberdeen Group, June 2006, p. 18-19

Aberdeen also provides the following general research findings regarding the ESB:

Issues:⁴

- “Existing application integration technology is too complex, resource-consuming, and slow-to-implement to keep up with business process changes.”
- “SOA technology from both application ISVs and development/middleware companies is the preferred technology base for solving the application integration problem.”
- “ESB is not the first of many successive SOA products to be deployed. It is often deployed with a suite of SOA middleware products.”

Key Business Value Findings:⁵

- “More than 90% of respondents are rapidly scaling the SOA adoption curve in 2006.”
- “ESB technology is complementing, not replacing, EAI technology.”
- “Enterprise ESB issues related to integration with registry/repository and scaling to high volumes are the greatest challenges ESB practitioners face.”

Implications and Analysis:⁶

- “The market is bifurcating into those who are using open standards but not SOA middleware products, and those mostly large companies who are seeking heavy-duty SOA middleware for mission-critical applications.”
- “SOA middleware is not a replacement for EAI or other technologies, but rather a supplement.”
- “Ease of integration flexibility with current and planned applications is the most frequently-mentioned buying criteria.”

Recommendations:⁷

- “Deliberately choose the SOA path your organization should take: SOA Lite, SOA ERP or Enterprise SOA.”
- “ESB technology is often packaged in an ESB suite, incorporating useful features for process management and orchestration, SOA governance, security, and message adaptation across legacy applications.”
- “SOA Lite is not a long-term, best-practices approach to maximizing business value.”
- “Third-party SOA services are a means to boost the learning curve.”

⁴ *Ibid*, p.1

⁵ *Ibid*, p. 5

⁶ *Ibid*, p. 11

⁷ *Ibid*, p. 18

PROBLEM

The State of Utah represents a confederation of “small companies” carrying out their respective IT missions. Shared services have been successful as a delivery model for utility based services such as network access, telephony, and some aspects of application hosting.

Diverse application development over time has fostered a complex and very costly environment for application development in State government. This environment is characterized by:

- high redundancies through isolated “stovepipe” solutions;
- missed business opportunities through lacking or inconsistent information, little exchange, and sometimes data ownership and sharing issues; and,
- high costs of maintenance modification and extension through unmanaged “spaghetti” interfaces.

Aberdeen identified the stumbling blocks listed in Figure 3 as primary issues in their survey that cause organizations to look toward SAO and ESB implementation.

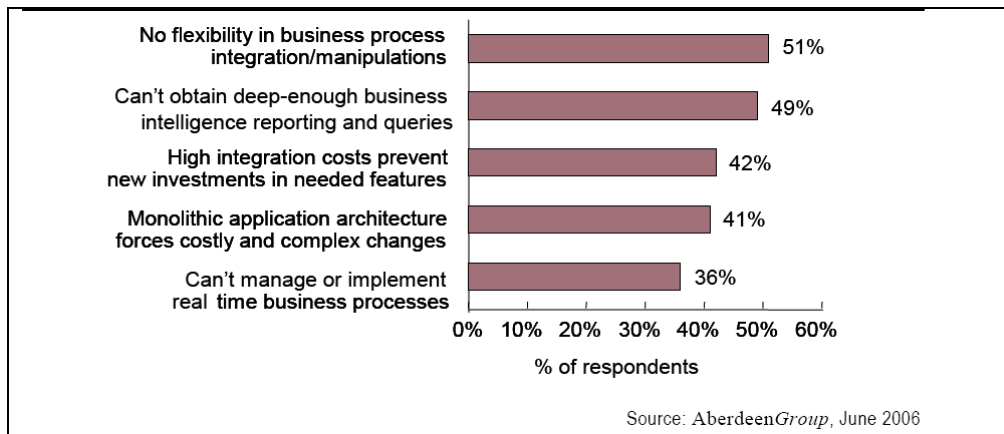


Figure 3. Application Integration Stumbling Blocks⁸

These issues appear to represent similar concerns within the State of Utah.

PREMISE

The premise of a Service Oriented Architecture (SOA) is that modular services can be assembled like Lego[®] pieces into new applications that leverage existing application and infrastructure investments. The principle challenge is that changes to business processes and business rules, sometimes called the process architecture, influence applications and the technical architecture in which they reside.

An ESB is fundamentally a messaging infrastructure that provides an abstraction layer on top of enterprise messaging systems to exploit the value of messaging without

⁸ *Ibid*, p. 2

writing code. The ESB provides an architecture that facilitates the task of integration of enterprise applications and services built on J2EE, .NET, C/C++, and other legacy environments within the reach of IT staff, using an event-driven service-oriented architecture.

The ESB is a software architecture construct, implemented by technologies found in a category of middleware infrastructure products usually based on Web services standards, which provides foundational services for more complex service-oriented architectures via an event-driven and XML-based messaging engine (the bus).

The ESB facilitates the ability of integration architects and developers to exploit the value of messaging without writing code. The foundation of an ESB is built on base functions broken up into their constituent parts, with distributed deployment where needed, as opposed to the more traditional EAI hub and spoke pattern.

ESB CHARACTERISTICS

ESB characteristics include:^{9, 10}

- ESB is based upon open standards for both the ESB solution components and the mechanisms for integrated applications to participate on the bus.
- Message based, ESB uses standard message notation, protocols, and transports.
- ESB can be distributed across a network for purposes of quality of service and other considerations.
- Routing, mediation, and invocation are the basic functions of an ESB.
- ESB facilitates interaction of resources and provides transactional support.
- An ESB must be reliable and guarantee message delivery.
- ESB requires the clear separation of message headers and message body.
- An ESB is usually operating system and language independent; it should work between Java and .Net applications, C++, and other legacy environments.
- ESB often uses XML and Web services to transport messages.
- ESB includes adapter standards (such as J2C/JCA) for incorporating existing applications into the bus.
- ESB includes support for asynchronous processing.
- ESB includes intelligent, content-based routing services.
- ESB includes a standardized security model to authorize, authenticate, and audit use of the ESB.
- ESB includes transformation services (such as XSLT) between the format of the sending application and the receiving application, including the transformation of data formats.
- ESB includes validation against schemas for sending and receiving messages.

⁹ Wikipedia. *Enterprise Service Bus*. http://en.wikipedia.org/wiki/Enterprise_service_bus

¹⁰ Michelson, Brenda, *Enterprise Service Bus Evaluation Framework: Criteria for Selecting an Enterprise Service Bus as an Integration Backbone*. Boston: Patricia Seybold Group, July 28, 2005

- ESB can uniformly apply business rules, enrichment of the message from other sources, splitting and combining of multiple messages, and the handling of exceptions.
- ESB can conditionally route or transform messages based on a central policy.
- ESB can be monitored for message latency and other characteristics described in a Service Level Agreement.
- ESB facilitates "service classes," responding appropriately to higher and lower priority users.
- ESB supports queuing, holding messages if applications are temporarily unavailable.
- ESB can handle a "publish and subscribe" messaging model, including event handling.
- ESB is comprised of selectively deployed application adapters in a (geographically) distributed environment.

KEY ESB BENEFITS

ESB IT Benefits:

- ESB facilitates enterprise integration.
- ESB serves as an infrastructure backbone for Service Oriented Architecture (SOA) applications and services, for both event driven and composite applications.
- ESB allows faster and cheaper accommodation of existing systems.
- With ESB, there is increased flexibility, making it easier to change as requirements change.
- ESB is standards-based.
- ESB scales from point solutions to enterprise-wide deployment (distributed bus).
- With ESB there is more emphasis on configuration rather than integration coding.
- ESB reduces time and effort to create new processes through the reuse of existing applications and data.
- There is increases flexibility, with ESB, to change complex system behavior by minimizing the hidden dependencies among applications, services, and middleware in a distributed environment.
- There is reduced TCO and greater flexibility, with ESB, accommodating future needs through the use of industry-standard interfaces and protocols.
- ESB reliably delivers messages across services, even after a software, network, or hardware failure.
- ESB provides a service hosting and management infrastructure that is highly distributed, yet centrally manageable.
- ESB disseminates information throughout the enterprise, as well as to customers and trading partners.
- ESB can be deployed incrementally, speeding delivery of service to customers and reducing risk for large, complex projects.

ESB Business Value:

- A standards-based ESB, combining new and existing technologies, enables a business to make use of comprehensive, flexible, and consistent approaches to integration while reducing the complexity of the applications being integrated.
- By using industry standard interfaces and protocols, there is a reduction in the cost and risk involved as business changes and new opportunities arise.
- Incrementally implementing an ESB, project by project, allows for better management of expenses while facilitating greater reuse of IT assets by separating application logic and integration tasks, providing a centrally managed, highly distributed, infrastructure.
- Services can be changed and added with minimal interruption to existing IT environments so that all parts of a business can react instantly to new information.
- Due to the complex and varying nature of business needs, the ESB unifies message oriented, event driven, and service oriented approaches for integrating applications and service.
- An ESB overcomes the differences in platform, software architecture, and network protocols, reducing the number, size, and complexity of integration interfaces.
- An ESB distributes data, reroutes, logs, and enriches information without rewriting applications.
- An ESB assures the delivery of transactions, even when systems and networks go off-line.

Aberdeen characterizes ESB adoption as accelerating rapidly, especially in large businesses. They predict that by the end of 2006, 90% of organizations that intend to adopt SOA and ESB will be at the programming stage, and well into adoption of SOA and ESB technologies.¹¹

Two governmental entities, Kentucky¹² and Washington, D.C.¹³, submitted NASCIO award applications in FY2006 that focused specifically on ESB projects. The State of Washington is also in the midst of a Mule[®] ESB implementation project. This, taken with significant ESB activity in the fortune 500 in 2006, would seem to be indicative of a more rapid rate of ESB adoption than might have been predicted in earlier assessments.

The State of Utah, like 55% of the customers in the Aberdeen survey, is looking at multiple platform and framework ESB implementation for both J2EE and .NET environments. Figure 4 identifies the drivers that Aberdeen identified for SOA ESB

¹¹ *Enterprise Service Bus and SOA Middleware*, Boston: Aberdeen Group, June 2006, p. 5

¹² Commonwealth of Kentucky., *NASCIO Recognition Award Application: Enterprise Service Bus*, June 2006 (Unpublished)

¹³ Washington DC., *NASCIO Recognition Award Application: Enterprise Architecture District Enterprise Integration Stack*, June 2006 (Unpublished)

adoption among the companies surveyed.¹⁴ BIC refers to Best in Class companies that are specifically defined as leaders from an SOA and ESB perspective.

From a State perspective, these drivers seem highly relevant. Alignment with the business and re-use of applications via Web services represent issues of substantial importance to the enterprise.

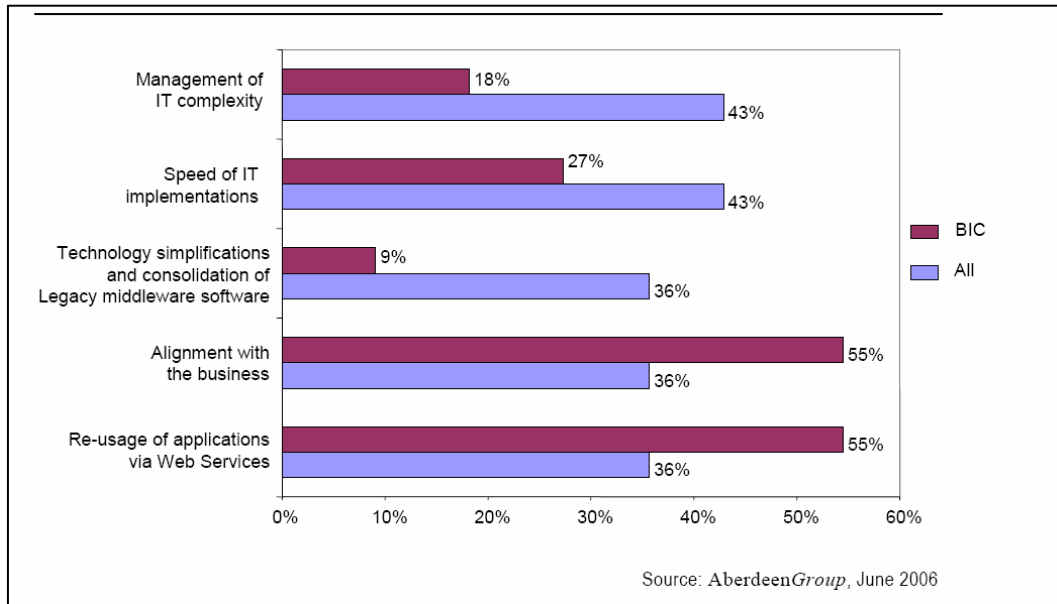


Figure 4. ESB Adoption Drivers

Other drivers identified get at the core issues of reducing complexity and decreasing the time to benefit for new and modified applications. Simplification has the potential for significant cost reductions and corresponding return on investment (ROI) for the State.

SOA/ESB GOVERNANCE

One of the most frequently identified critical success factors with SOA and ESB implementation is tactical governance. Governance is critical to the State's success with SOA. Governance involves establishing responsibilities and empowering employees, whereas management involves assuring that the governance policies are actually implemented. Technology can be used to perform management. Governance that is managed during service invocation can be effectively managed by an ESB, simplifying the responsibilities of both the provider and consumer.

SOA governance has many aspects, including service:¹⁵

- definition, including the scope, interface, and boundaries of a service;
- deployment lifecycle, including lifecycle stages;
- versioning, including compatibility;
- migration, including deprecation and sunseting;

¹⁴ *Enterprise Service Bus and SOA Middleware*, Boston: Aberdeen Group, June 2006, p. 3

¹⁵ Wolf, Barry. *Introduction to SOA Governance*, IBM, June 13, 2006.

- registries, including dependencies;
- message model, including canonical data models;
- monitoring, including problem determination;
- ownership, including agency organization;
- testing, including duplicated testing; and,
- security, including ranges of acceptable protection.

As the State moves toward SOA and ESB implementation, the governance and related management issues have to be addressed or the implementation will likely be somewhat chaotic and will not meet the service, interoperability, or reliability required by agencies.

REALITY

Notwithstanding the evident benefits of an ESB, the technology has been slow to catch on in many sectors. Gartner® has characterized worldwide use of application integration as follows:

“...barely more than 10 percent of all application development projects use an integration suite, and most projects (60 percent) use point solutions for integration, such as database management system gateways, adapter tools, electronic data interchange tools, screen scrapers, integration servers, transformation engines, extraction, transformation and loading tools, enterprise information integration tools, transaction delivery networks or enterprise service buses (ESBs). The remaining 30 percent of projects use no technology specifically aimed at integration. Rather, they use only basic programming languages, platforms, tools and middleware, such as application servers, message-oriented middleware and file transfer utilities.”¹⁶

This characterization is descriptive of the current application integration environment of the State of Utah. Since this statement from Gartner was released, ESB adoption trends appear to be accelerating, especially with the advent of credible open source standards-based ESB products. Aberdeen notes that over 90% of all respondents will end the year with experience in SOA planning, design, or programming.¹⁷

Aberdeen also identified, in Figure 5¹⁸, the obstacles that are getting in the way of ESB implementation and acceptance.

¹⁶ Correia, Joanne M, et al., *Forecast: AIM and Portal Software, Worldwide, 2005-2010 (Executive Summary)*, March 2006, Gartner Research G00138499, March 22, 2006.

¹⁷ *Enterprise Service Bus and SOA Middleware*, Boston: Aberdeen Group, June 2006, p. 5

¹⁸ *Ibid*, p. 9

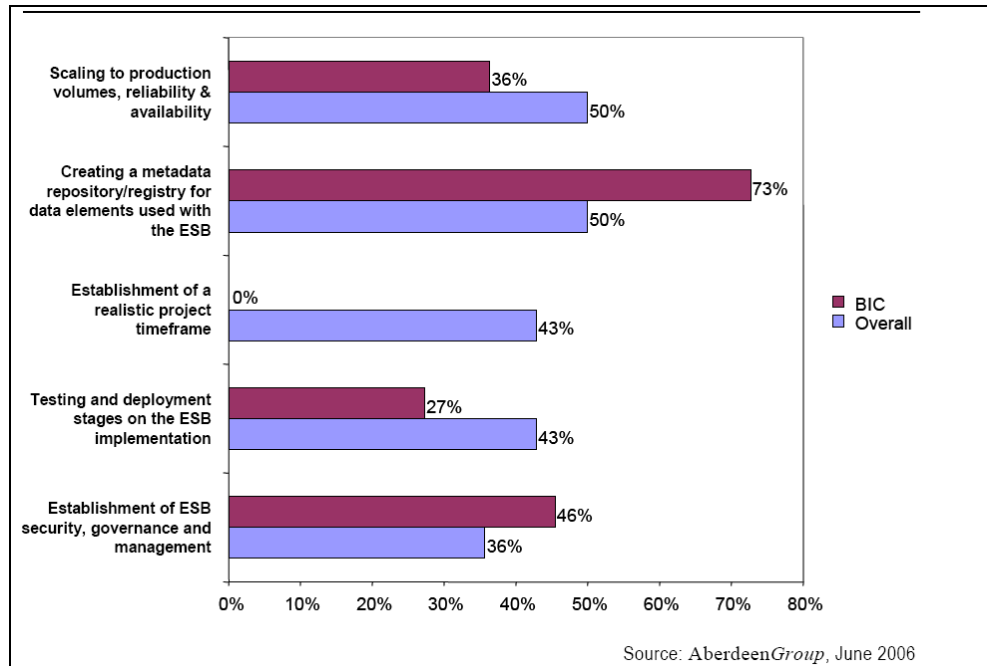


Figure 5. ESB Technology Stumbling Blocks

ALTERNATIVE SOLUTIONS

Since ESB technologies were ignored by nearly half of the Aberdeen survey pool, it seems reasonable to consider what other alternatives are being employed. Strategies most commonly considered as alternatives to an ESB approach include:¹⁹

- Business Process Integration or Management (BPM) 64%
- Enterprise Application Integration Software (EAI) 50%
- Message Oriented Middleware (MOM) 43%
- IBM® Message Queuing (MQ) 43%
- CORBA Object Request Broker (ORB) 29%

Many of these solutions offer some degree of functional equivalency to ESB functionality, but with the exception of MQ, none of these approaches represent a significant availability of infrastructure within the State. All of these approaches are heavily fragmented with agency specific deployments.

Aberdeen suggests that large companies are “clearly voting with their wallets to buy and deploy ESBs and they are most likely to have more than a year’s experience in programming with SOA technology.”²⁰ A number of State agencies also have this base of experience with SOA and Web services. This being the case, Aberdeen suggests criteria for making an ESB purchase decision as listed in Table 1.

¹⁹ *Ibid*, p. 8

²⁰ *Ibid*, p. 9

Weighted importance of specific drivers for the ESB technology decision			
	Very Important	Somewhat Important	Not Important
Ease of integration flexibility with current and planned applications	43%	25%	0%
ESB business process control, change, management, governance, and life-cycle features	29%	35%	6%
Completeness of vendor's SOA product offering	20%	43%	4%
Completeness of vendor's ESB product offering (e.g., suite)	20%	43%	4%
ESB security features	25%	37%	6%
ESB features protect our organization's legacy middleware investments	29%	31%	8%
ESB scalability, robustness, reliability, clustering, and fail-over features	25%	35%	4%
ESB process modeling and BPEL capabilities	20%	39%	8%
Extensive range of ESB communications connectors and transport options	22%	35%	12%
ESB business process orchestration capabilities	22%	35%	8%
ESB compliance with industry standards	12%	47%	6%
Proven ability of ESB to sustain high volumes in production	20%	35%	12%
ESB mediation capabilities	18%	37%	8%
ESB development environment	20%	33%	10%
ESB is tightly integrated with vendor's other SOA technology	22%	29%	10%
ESB is tightly integrated with vendor's ERP applications	14%	35%	12%

Source: AberdeenGroup, June 2006

Table 1. Factors in ESB Purchase Decision²¹²¹ *Ibid*, p. 14

EVALUATION SCOPE AND ARCHITECTURAL PREMISE

ESB products represent a range of architectural approaches and premises. The focus of this comparison is a detailed comparison of three of the products that are most likely to be deployed as an ESB for the State of Utah among the following alternatives:

Enterprise SOA Products: Architected and deployed for mission-critical, high volume applications and scalability; usually standards based. Many integration adapters are available, lowering legacy integration costs. Vendor support and some level of training is available.

- **Commercial and Open Source Integration/Object Broker ESB Products**

- Mule*
- Fiorano ESB®
- Cape Clear®
- Progress (Sonic®)
- Tibco Active Enterprise®
- Iona Artix ESB®

*Mule can also be deployed as an SOA Lite product but has a sufficient number of mission critical large scale production deployments to be considered in this category.

- **Service Component Architecture (SCA) ESB Products**

Defines ways to describe service models and interaction. Java, C++, and BPEL are also supported. Does not usually deal with transport, except with other add-on products.

- BEA AquaLogic Suite®
- IBM Web Sphere ESB®
- Oracle Application Server ESB®

SOA Lite Products: Pure standards-based open source. Wide adoption: basic skills are easily obtainable. Limited or no major mission critical and high volume deployments are available.

- **JBK Based ESB Products**

Integration container API with specific focus on Java.

- Apache ServiceMix®
- Celtix®
- Sun GlassFish®
- JBoss®

- **Web Service Based ESB Products**

Uses ESBs from WS specifications with WSDL endpoints, and always uses SOAP as the message payload.

- Apache Synapse®
- Blue Titan Network Director®

From an architectural premise perspective²² this report considers that the following are ESB architectural premises for the State, and enterprise vendor products should be in alignment to be successful:

- The State of Utah future will continue to include heterogeneous applications, information, and services. J2EE, C/C++, and .NET environments will all be in use by agencies.
- The State foresees future application development and integration services built around Web Services specifications.
- The State environment incorporates an RPC (verb) view of services. Services provide action, via an operation.
- The State environment includes a REST (noun) view of services. Services deliver and manage documents.
- Business and data mediation challenges are emphasized over technical mediation challenges.
- The State expects the ESB to be intelligent and assumes that intelligence will also exist at the endpoints.
- The human interface (user interface) is a type of integration service that performs actions in many integration scenarios.
- ESB implementation must add value and capability to existing infrastructure and may not have a significant architectural impact on existing deployed architectures.

The products selected for a suggested detailed scenario or use-based comparisons for this report were as follows:

- Mule (Open Source—Open Standards Based Architecture
- Apache ServiceMix—Open Source—JB I Architecture, or another To Be Determined Commercial Product.
- BEA AquaLogic—Commercial—SCA Architecture

Data for the comparisons in this report was gathered from published documentation from the vendor, ESB providers, and from external sources such as Forrester®, Gartner, and the Patricia Seybold Group®. Accuracy of the information in the comparative matrix is based solely upon information provided by the vendors of the respective products.

²² Michelson, Brenda, "SOA and Integration Infrastructure, Understand the Provider's Intent (architectural premise)" *Elemental Links*, August 2, 2005.

The evaluation criteria were gathered based upon a review of all of the pertinent vendor documentation and specific recommendations from the Patricia Seybold Group.²³ The criteria were then reviewed by assigned State IT developers with experience in this area to assure relevance to the State development environment. Documentation references are cited in the footnotes and the references section of this document.

²³ Michelson, Brenda, *Enterprise Service Bus Evaluation Framework: Criteria for Selecting an Enterprise Service Bus as an Integration Backbone*. Boston: Patricia Seybold Group, July 28, 2005.

PRODUCT COMPARISON INFORMATION

Product comparisons are presented on a variety of levels of analysis. Figure 6 is taken from the Forrester study²⁴ of commercially available ESB products completed in the second quarter of 2006. The categories are very general and do not lend themselves to detailed functional comparisons, but are useful in identifying the most highly rated commercial solutions. No open source offerings are included in this comparison.

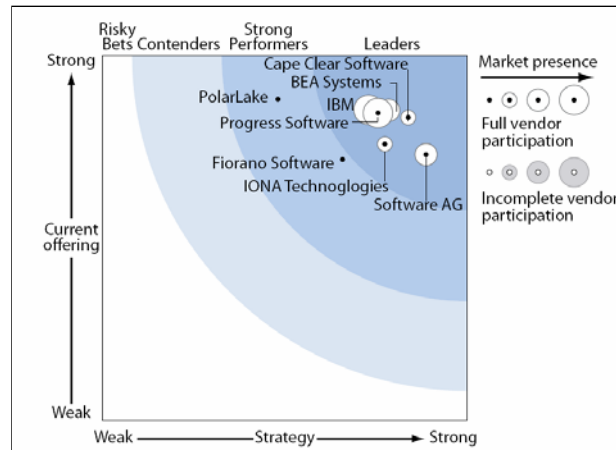


Figure 6. Forrester Research, Inc. Summary Evaluation

Forrester concluded that “Cape Clear Software and BEA Systems were the top two performers overall.”²⁵ This evaluation covered commercially available ESB products from BEA®, Cape Clear, Fiorano, IBM, Iona Technologies®, Polar Lake®, Progress (Sonic), and Software AG®. The study compared ESB products on the basis of the following general evaluation criteria²⁶:

Current Offering

- **Connection:** How sophisticated is the ESB support for messaging and connectivity?
- **Mediation:** How rich of a set of mediation services and capabilities does the product provide?
- **Control and Change:** What capabilities does the ESB provide in the area of control and change management?

²⁴ Vollmer, Ken, and Gilpin, Mike, *The Forrester Wave™: Enterprise Service Bus, Q2 2006*, Forrester Research, Inc., June 30, 2006.

²⁵ *Ibid*, p. 1.

²⁶ *Ibid*, p. 7

Strategy

- **Product Strategy and Vision:** How strong is the vendor's product strategy and vision?
- **Strategic Alliances:** How strong are the vendor's strategic alliances?
- **Corporate Strategy:** How strong is the vendor's corporate strategy?
- **Solution Cost:** What is the relative cost of the vendor's ESB solution?

Market Presence

- **Customer Base:** How large is the vendor's installed base of customers for this product and all products?
- **New Customers:** How many customers are buying or upgrading any version of this product?
- **Delivery Footprint:** What is the vendor's method of delivery?
- **Financial Viability:** How strong is the vendor's financial position?

	Forrester's Weighting	BEA Systems	Cape Clear Software	Fiorano Software	IBM	IONA Technologies	PolarLake	Progress Software	Software AG
CURRENT OFFERING	50%	4.23	4.10	3.57	4.26	3.75	4.38	4.21	3.62
Connection	30%	2.95	4.03	3.48	3.98	4.43	3.70	4.35	2.50
Mediation	40%	4.76	3.55	3.04	4.07	3.25	4.62	3.88	4.15
Control and change	30%	4.80	4.90	4.39	4.80	3.76	4.74	4.50	4.05
STRATEGY	50%	3.92	4.20	3.28	3.68	3.88	2.39	3.80	4.44
Product strategy and vision	30%	5.00	4.60	3.00	5.00	3.80	3.40	4.60	4.60
Strategic alliances	20%	4.20	5.00	2.80	5.00	2.20	2.20	4.20	2.80
Corporate strategy	10%	5.00	5.00	3.00	5.00	5.00	2.10	5.00	5.00
Solution cost	40%	2.70	3.30	3.80	1.70	4.50	1.80	2.70	5.00
MARKET PRESENCE	0%	3.60	2.60	1.43	4.84	2.40	0.97	4.36	3.58
Installed base	20%	2.60	3.40	1.80	4.20	1.80	1.00	3.40	3.40
New customers	45%	3.40	2.20	0.00	5.00	1.80	0.00	5.00	3.00
Delivery footprint	20%	4.00	2.40	1.60	5.00	3.00	1.60	3.40	4.00
Financial viability	15%	5.00	3.00	5.00	5.00	4.20	3.00	5.00	5.00

All scores are based on a scale of 0 (weak) to 5 (strong).

Source: Forrester Research, Inc.

Table 2. Forrester Research, Inc. ESB Scoring Results

To get a useful comparison on the same level of detail as represented in Table 2, the Forrester criteria were applied to open source products Apache ServiceMix and Mule. The results of these evaluations are included in Table 3. This provides a basis for comparison between the commercial products that Forrester reviewed and comparable leading open source products.

The process of adding open source vendors to the Forrester matrix required a number of direct contacts with the vendors as well as an analysis of available public and restricted documentation sources. While it was not completely possible to replicate all of the detailed Forrester criteria, the result is consistent with the Forrester methodology and provided some useful comparable assessment.

		Apache	BEA Systems	Cape Clear	Fiorana	IBM	Iona	Mule	Polar Lake	Progress (Sonic)	Software AG
CURRENT OFFERING	50%	4.00	4.17	4.16	3.64	4.28	3.81	4.13	4.35	4.24	3.57
Connection	30%	4.00	2.95	4.03	3.48	3.98	4.43	4.20	3.70	4.35	2.50
Mediation	40%	4.00	4.76	3.55	3.04	4.07	3.25	4.20	4.62	3.88	4.15
Control and change	30%	4.00	4.80	4.90	4.39	4.80	3.76	4.00	4.74	4.50	4.05
STRATEGY	50%	4.38	4.23	4.48	3.15	4.18	3.88	4.63	2.38	4.13	4.35
Product strategy and vision	30%	4.00	5.00	4.60	3.00	5.00	3.80	5.00	3.40	4.60	4.60
Strategic Alliances	20%	4.50	4.20	5.00	2.80	5.00	2.20	4.50	2.20	4.20	2.80
Corporate strategy	10%	4.00	5.00	5.00	3.00	5.00	5.00	4.00	2.10	5.00	5.00
Solution cost	40%	5.00	2.70	3.30	3.80	1.70	4.50	5.00	1.80	2.70	5.00
MARKET PRESENCE	0%	4.13	3.60	2.60	1.43	4.84	2.40	4.50	0.97	4.36	3.58
Installed base	20%	5.00	2.60	3.40	1.80	4.20	1.80	4.50	1.00	3.40	3.40
New customers	45%	3.00	3.40	2.20	0.00	5.00	1.80	4.50	0.00	5.00	3.00
Delivery footprint	20%	4.50	4.00	2.40	1.60	5.00	3.00	5.00	1.60	3.40	4.00
Financial viability	15%	4.00	5.00	3.00	5.00	5.00	4.20	4.00	3.00	5.00	5.00
Unweighted Average Score		4.18	4.04	3.76	2.90	4.43	3.43	4.45	2.56	4.18	3.95

All scores are based on a scale of 0 (weak) to 5 (strong)

Table 3. Consolidated ESB Scoring Results Based Upon the Forrester Criteria

VENDOR PROFILES

Vendor profile information²⁷ is quoted from the Forrester evaluation, with the addition of profiles for Apache and Mule based on information provided on their respective Web sites.

- Apache:** “Apache ServiceMix is an Enterprise Service Bus (ESB) that combines the functionality of a Service Oriented Architecture (SOA) and an Event Driven Architecture (EDA) to create an agile, enterprise ESB. ServiceMix is lightweight and easily embeddable, has integrated Spring support and can be run at the edge of the network (inside a client or server), as a standalone ESB provider or as a service within another ESB. You can use ServiceMix in Java SE or a Java EE application server. ServiceMix is an open source distributed ESB built from the ground up on the Java Business Integration (JBI) specification JSR 208 and released under the Apache license.”²⁸

²⁷ Ibid, p. 10-11.

²⁸ Apache ServiceMix Web Site <http://incubator.apache.org/servicemix/main/home.html>

- **BEA Systems:** "As a longtime leader in the application platform market, BEA has provided application integration for years (WebLogic Integration). Last year, it entered the ESB market with a new ESB: AquaLogic Service Bus, which was intended to enable service-oriented integration across all platforms, not just centered on WebLogic. Some dependencies on the WebLogic runtime remain, but they are to be reduced in future releases. The AquaLogic Service Bus provides solid ESB capabilities but does not include service orchestration, which BEA delivers in WebLogic Integration."
- **Cape Clear:** "One of the early innovators in the ESB market, Cape Clear has grown its offering to a broad suite by adding service orchestration and some management features. It now has one of the deepest implementations available of the Web services stack. Cape Clear is also known for the productivity its tools bring to SOA development. It is a small, privately held company, but it has built a greater market presence than would be expected for its size."
- **Fiorano Software:** "Another early market entrant, Fiorano, like Sonic Software, built its ESB on its JMS product, FioranoMQ. Its service orchestration tools are strong, as is its service life-cycle management. But Fiorano is a small privately held company, with limited market presence."
- **IBM:** "IBM has recently entered the ESB market with one new and two updated products announced in September 2005: WebSphere Enterprise Service Bus (new), WebSphere Message Broker v6.0 (updated), and WebSphere Process Server v6.0 (updated). The vendor refers to these three products as the IBM SOA Foundation, and together they provide comprehensive ESB support. IBM also provides an ESB appliance, but SOA hardware appliances were beyond the scope of this evaluation."
- **IONA Technologies:** "IONA, another longtime player in the middleware market, entered the ESB market in 2004. It has made its mark at a number of customer sites, although its ESB Artix represents a small but growing proportion of its business. The company has done a good job of building on its architectural advantages to establish a niche position at the high end of the market. The vendor recently released Artix 4.0, which remedies a number of its predecessor's shortcomings."
- **Mule:** "Mule is the leading open source ESB and integration platform. It is a scalable, highly distributable object broker that can seamlessly handle interactions with services and applications using disparate transport and messaging technologies. Mule is a light-weight messaging framework. It is a highly distributable object broker that can seamlessly handle interactions with other applications using disparate technologies, transports and protocols. Mule leverages many open source projects such as Axis, Spring, ActiveMQ, Plexus and PicoContainer. Mule fills a void in enterprise java development where an

application requires complex interactions with a variety of systems on a variety of platforms.”²⁹

- **PolarLake:** “Another company that has been in the integration business for years, PolarLake made the switch to an SOA-driven strategy much earlier than many of its counterparts. It has not, however, implemented the same depth of Web services support that can be obtained from the leading vendors. PolarLake’s emphasis is on productivity and maintainability of integrated systems by providing tools that enable all integration tasks to be performed without programming. It also provides rich support for data transformation and process modeling and has recently added support for UDDI.”
- **Progress (Sonic):** “Sonic Software was formerly an independent operating unit of Progress Software, but it has recently been incorporated into the parent company’s infrastructure. Sonic Software was there when the ESB market began and was responsible for a large part of the early growth in the market. It continues to grow, having recently acquired Actional Software. Release 7.0 of this ESB combines the features of the previous Sonic ESB v6.1 with key features acquired from Actional into an integrated ESB stack that provides several improved features at a lower price.”
- **Software AG:** “This is the first time that Forrester has evaluated Software AG in this category, and our evaluation found that the vendor provides a strong ESB product that is particularly notable in its bundled CentraSite registry/repository. Software AG also provides strong capabilities for mainframe integration, especially for its existing customers.”

Other comparisons of ESB suites without reference to open source solutions are available from NWC Real World Labs³⁰. These eight evaluations added some interesting category information but did not add substantial additional value. The principle evaluation criteria for this report included:

- core bus features (routing, transformation and mapping, orchestration, protocol support, and management/configuration);
- integration (adapter support, Web services, and management/configuration); and,
- price.

The results did not provide sufficient detail to permit a more detailed analysis.

²⁹ Mule Web Site <http://mule.mulesource.org/wiki/display/MULE/Overview>

³⁰ Real World Labs Report Card: ESB Suites at <http://i.cmpnet.com/nc/1705/graphics/1705f2report.gif> 2006.

ARCHITECTURE PREMISE ALIGNMENT

Architectural premise alignment was suggested in a blog³¹ by Brenda Michelson as a key factor in determining how well a particular vendor's ESB framework fits with the needs of the customer buying the product.

From an architectural perspective, Michelson suggests that an ESB decision hinges on some of the larger questions that should be part of a business driven architecture decision, such as: "What problems are businesses trying to solve? What are the business and technology opportunities? How do they envision the future of information interchange? What will future IT portfolios look like? How service-oriented can we (should we) be? What new business and technology problems/challenges/opportunities will the solutions create?"

She suggests that there is a need to know more than the product offering and how it works. There is a need to understand the overall context, and more specifically, the intent of the vendor of the product. This understanding is what is referred to as "architectural premise." Understanding the premises that are associated with potential use of an ESB by agencies helps the State align architectural vision with that of the provider.

The statements in Table 4 are answers from a State perspective to questions suggested by Michelson for consideration of the integration provider's view of the challenges of integration and the future of services. The statements are worded as affirmative answers that are relevant to the complex and diverse architecture of the State.

³¹ Michelson, Brenda M., *Elemental Links: The Home of Business-Driven Architecture – a Weblog*, http://elementallinks.typepad.com/bmichelson/2005/08/soa_and_integra.html

	Alignment			
	Other	BEA	Mule	IBM
Architectural Premise				
The State of Utah future will continue to include heterogeneous applications, information, and services. J2EE, C/C++, and .NET environments will all be in use by agencies.				
The State views future application development and integration services built around Web Services specifications.				
The State environment incorporates an RPC (verb) view of services. Services provide action, via an operation.				
The State environment includes a REST (noun) view of services. Services deliver and manage documents.				
Business and data mediation challenges are emphasized over technical mediation challenges.				
The State expects the ESB to be intelligent, and assumes that intelligence will also exist at the endpoints.				
The human (user interface) is a type of integration service and they perform actions in many integration scenarios.				
ESB implementation must add value and capability to existing infrastructure and may not have a significant architectural impact on existing deployed architectures.				

Table 4. Architectural Premise Alignment with Detailed Evaluation Products

The diversity of the State architecture and the multiple demands for different types of ESB services speak loudly in favor of solutions that have a great deal of flexibility and can be quickly deployed as both centralized and endpoint infrastructure. The ideal ESB environment for the State would appear to be one that supports diverse protocols and application server environments, is based upon open standards, is not unduly tied to specific service components or objects, and must be able to add value to the environment without unduly impacting existing deployments. ESB implementation must be cost effective and, at least in the current State environment, must not force compliance or utilization. A State ESB implementation must be capable of serving as a central resource, but is more likely to be deployed at the edge in support of specific agency business requirements.

DEPARTMENT OF TECHNOLOGY SERVICES ESB COMPARISONS

Based upon all of the published comparisons of ESB products, a set of criteria were established that would provide a more granular look at some of the more likely ESB applications that could be selected. The criteria included both general application and key technical feature assessments. The criteria were scored by vendors and DTS staff based upon the following scoring criteria:

- Range: 1 (Lowest) to 5 (Highest)
 - 4 to 5 A = Acceptable (fully meets requirement)
 - 2 to 3 PA = Potentially Acceptable (partially meets requirement)
 - 0 to 1 U = Unacceptable (does not meet requirement)

The scoring methodology is consistent with RFP scoring methods. It is important to note that there is no expectation that any single vendor product will meet all of the technology functionality in Table 5. Breadth of response and score is indicative of the ability of the vendor to incorporate many core technologies and development methodologies. The scenario based evaluation recommended in this report in Appendix A, is designed to help establish which of the many functionalities in Table 5 are really essential to the State ESB implementation.

The criteria chosen for comparison and the scores are included in Table 5 and its subparts, as follows:

Table 5 Part 1: ESB Evaluation Comparison—Business Drivers

Enterprise Service Bus Product Comparison as of 10/17/2006

		Mule 1.3	BEA AquaLogic	Apache ServiceMix	Fiorano	IBM Web Sphere ESB
	EVALUATION CRITERIA/CHARACTERISTICS					
Item #	Business Driver Alignment Assessment					
1.01	Ease of integration flexibility with current and planned applications	5	4	3	4	3
1.02	ESB business process control, change, management, governance, and life-cycle features	3	4	1	5	4
1.03	Completeness of the ESB product offering	4	5	2	4	4
1.04	ESB security features and functionality	4	4	3	5	4
1.05	ESB features protect the State's legacy middleware investments	4	4	2	4	3
1.06	ESB scalability, robustness, reliability, clustering, and fail-over features	3	5	2	4	4
1.07	ESB process modeling with BPEL capabilities	1	4	2	5	4
1.08	Extensive range of ESB communications connectors and transport options	5	5	2	5	3
1.09	ESB business process orchestration capabilities	4	5	2	5	5
1.10	ESB compliance with industry standards	5	5	5	4	5
1.11	Proven ability of ESB to sustain high volumes in production	5	5	2	5	5
1.12	ESB mediation capabilities	5	5	2	5	5
1.13	ESB development environment flexibility	4	4	4	5	4
1.14	ESB integration with other vendor SOA technologies	5	4	2	4	2

Table 5 Part 2: ESB Evaluation Comparison—Other General Criteria

Item #	Deployment Topology					
1.16	Client/Server	5	0		0	
1.17	Enterprise Service Network (ESN)	5	0		5	
1.18	ESB	5	5		5	5
1.19	Peer to Peer	4	0		5	
1.20	Remote deployment and management	3	5	0	5	3
Item #	Operating System Deployment Options					
1.21	Mac OSX	3	0	1	5	0
1.22	Red Hat Linux	5	5	5	5	5
1.23	Solaris SPARC/x86	5	5	4	5	5
1.24	Suse Linux	5	5	4	5	5
1.25	Windows Server	4	5	2	5	4
Item #	Deployment Complexity					
1.26	Impact on existing infrastructure	5	4	3	5	1
1.27	J2EE Application Server Installation	3	5	2	0	2
1.28	Independent server installation	5	5	3	5	2
Item #	Support Options					
1.29	24X7 Support Availability	5	5	2	5	5
1.30	Contract Support Availability	5	5	2	5	5
1.31	Custom Engineering Services	3	4	2	5	5
Item #	License and Support Costs					
1.32	License cost (Specify Method)	OSS	CPU	OSS	CPU	CPU
1.33	Annual Support Cost	None	% List	Fixed	% List	% List
1.34	Dependencies on other Product Components	5	4	4	5	2
Item #	Installed Customer Base					
1.35	Private Sector	100	60	5	400	
1.36	Public Sector	30	10	3	50	
Item #	Quality of Service, Monitoring & Lifecycle Support					
1.37	Service Level Agreement Support	3	5	2	4	4
1.38	Monitoring and Management	4	4	4	5	4
1.39	Integrated monitoring, tracing, and logging	4	4	2	4	4
1.40	Eclipse functionality	3	4	2	4	3
1.41	Service Lifecycle management including development, reuse, integration, deployment, management, and optimization	4	4	2	4	4
Section Point Total		93	83	46	96	68

Table 5 Part 3: ESB Evaluation Comparison—Technology Components

		Mule 1.3	BEA AquaLogic	Apache ServiceMix	Fiorano	IBM Web Sphere ESB
TECHNOLOGY COMPONENT EVALUATION						
Item #	Java					
2.00	1.4/1.5/1.6	5	5	3	5	3
Item #	API					
2.01	REST	5	4	4	4	2
2.02	Proprietary	5	5	2	0	5
Item #	End to end event support					
2.03	Routing	5	5	3	5	4
2.04	Transport	5	5	2	5	5
2.05	Transformation	5	5	3	5	4
Item #	Service registry and metadata management					
2.06	UDDI V3 or greater	0	5	0	5	3
Item #	Application Server Support					
2.07	Apache Tomcat	5	2	4	5	2
2.08	Geronimo	3	1	4	5	2
2.09	Jboss	5	2	1	5	2
2.10	Jetty	5	1	1	5	2
2.11	Jrun	4	1	1	5	2
2.12	Oracle	5	1	1	5	4
2.13	Resin	4	1	1	5	2
2.14	Web Sphere	5	2	1	5	5
2.15	WebLogic	5	5	1	5	1
Item #	Transport					
2.16	Supports synchronous, asynchronous and request response events	5	5	2	5	5
Item #	Integration/Framework					
2.17	EJB	5	5	1	5	5
2.18	GigaSpaces	5	5	1	0	
2.19	HiveMind	5	1	1	0	1
2.20	JavaSpaces	5	3	1	0	1
2.21	JB1	4	3	5	0	4
2.22	JCA	5	5	5	5	4
2.23	JNDI	5	5	0	5	1
2.24	JOTM	5	0	0	0	1
2.25	JTA	5	5	0	5	1
2.26	PicoContainer	5	0	0	0	0
2.27	Plexus	5	0	0	0	0
2.28	Spring	5	5	3	5	3
2.29	Struts	5	4	3	0	3

Table 5 Part 4: ESB Evaluation Comparison—Technology Components

		Mule 1.3	BEA AquaLogic	Apache ServiceMix	Fiorano	IBM Web Sphere ESB
Item #	Development Tools					
2.30	Component development environment for writing intelligent adapters in multiple languages	4	5	3	5	5
2.31	Developers insulated from messaging layer	5	5	5	5	5
2.32	Documented Service API for developing new services	5	5	4	5	5
2.33	JMS compliant messaging API	5	5	5	5	5
2.34	Open platform for 3rd party tools, IDEs, etc.	5	5	5	5	2
2.35	Standards based OS agnostic	5	5	5	5	2
2.36	Supports full XML standard	5	5	3	5	5
Item #	Web Services					
2.37	Axis	5	2	5	4	2
2.38	REST	5	3		4	3
2.39	SOAP	5	5	5	4	5
2.40	WebMethods Glue	5	0		4	1
2.41	Xfire	5	0	5	0	0
Item #	Security					
2.42	ACEGI	5	0	0	0	0
2.43	JAAS	5	5	0	4	0
2.44	PGP	5	3	2	4	3
Item #	Other Technology Support					
2.45	BPEL	3	5	2	5	5
2.46	jBPM	5	0	2	0	0
2.47	JSR -223 (Scripting)	5	5	5	0	5
2.48	OGNL Filters	5	0	0	0	0
2.49	Quartz	5	0	5	0	0
Section Point Total		97	63	61	54	53
Total Points All Sections		392	309	198	322	255

The overall evaluation was based upon a point distribution that was structured around the commonly used State RFP criteria for evaluation, with some modifications for this specific product type. Pricing, for example, is usually set at a 30% threshold, but was set lower since some of the products reviewed were open source and did not have a fixed upfront licensing cost.

Evaluation Weighting	Points	Percent
Business Driver Alignment	75	10%
Deployment, Monitoring, QOS, and Lifecycle	120	16%
Technology Component Capability	250	34%
Price	150	20%
Installed Base and Reference Accounts	75	10%
Financial Viability of the Vendor and Product	75	10%
TOTALS	745	100%

Table 6: Composite Product Evaluation Weighting

Financial viability scores were taken from other published studies previously cited by Forrester.³² Estimates for financial viability were used as cited in Table 3 for all vendors. Installed base and reference scores were based upon information supplied by the vendors or directly available from published sources. The table that follows provides summary point totals for vendors for each of the weighted product evaluation sections:

	Mule 1.3	BEA AquaLogic	Apache ServiceMix	Fiorano	IBM Web Sphere ESB
EVALUATION SUMMARY					
Business Driver Alignment	62	67	37	68	57
Deployment, Monitoring, QOS, and Lifecycle	93	83	46	96	68
Technology Component Capability	237	159	115	158	130
Price	120	50	150	50	50
Installed Base and Reference Accounts	60	60	15	75	75
Financial Viability of the Vendor and Product	45	75	60	75	75
TOTAL Points	617	494	423	522	455
Average Score (0-5)	4.14	3.32	2.84	3.50	3.05

Table 7: Composite Product Evaluation Scores

All of the vendors analyzed provide product offerings that could potentially be suitable for State use. Some of the technology functions are clearly of greater importance than others, and those would need to be identified and assessed for actual suitability to the needs of the State. The evaluation criteria listed in Appendix A move in that direction based upon actual tests of products. All of the vendors with the exception of Apache ServiceMix seem to have a sufficient installed base for reference and real world technology assessment.

FINANCIAL ANALYSIS AND COST RECOVERY

Cost estimates for each product were based upon a comparable level of effort assumption for initial deployment and ongoing use. Those assumptions need to be verified by applying the evaluation criteria in Appendix A. Some of the solutions have proprietary components and may require greater effort for integration and ongoing support.

Training cost estimates are based upon training requirements for the relative complexity of each product, and are based on training for six senior level IT Analyst 3 positions in DTS. The assumption would be that these personnel would serve as resource for training other analysts.

This analysis looks at initial procurement and ongoing support costs for each vendor in the comparison and year two and three ongoing cost estimates. Table 8 lists the cost

³² Vollmer, Ken, and Gilpin, Mike, *The Forrester Wave™: Enterprise Service Bus, Q2 2006*, Forrester Research, Inc., June 30, 2006.

information as supplied by vendors for licensing and support for three environments with two servers in each environment for Production, Acceptance Testing, and Development. Server cost estimates were provided by DTS staff. All cost estimates are based upon a three year cost forecast and assume additional server, licensing, and support resource requirements for each year. Actual server deployment costs and configurations would probably be somewhat lower than the estimated costs.

Year 1 Cost Analysis						
Vendor	Cost Per License (1)	Maintenance & Support Cost (1)	Total CPU License Cost	Total Annual Maintenance & Support	Initial Server Costs (2)	Initial Training Costs (3)
Apache Service Mix	\$ -	\$ -	\$ -	\$ -	\$ 48,000	\$ 18,240
Total Cost	\$ -	\$ -	\$ -	\$ -	\$ 48,000	\$ 18,240
AquaLogic Service Bus	\$ 19,250	\$ 7,350	\$ 67,375	\$ 14,700	\$ 48,000	\$ 18,240
AquaLogic Data Service Platform	\$ 26,400	\$ 10,080	\$ 92,400	\$ 20,160		\$ 9,120
AquaLogic Service Registry	\$ 57,750	\$ 22,050	\$ 202,125	\$ 44,100		\$ 9,120
AquaLogic BPM	\$ 57,500	\$ 24,150	\$ 201,250	\$ 48,300		\$ 9,120
AquaLogic User Interaction	\$ 41,250	\$ 15,750	\$ 144,375	\$ 31,500		\$ 9,120
WebLogic Integration	\$ 34,100	\$ 13,020	\$ 119,350	\$ 26,040		\$ 9,120
Total Cost*	\$ 236,250	\$ 92,400	\$ 826,875	\$ 184,800	\$ 48,000	\$ 63,840
Fiorano ESB	\$ 28,000	\$ 8,000	\$ 128,000	\$ 48,000	\$ 48,000	\$ 36,000
Fiorano Components and Adapters	\$ 21,000	\$ 5,250	\$ 157,500	\$ 31,500		
Other Costs	\$ 40,000					
Total Cost	\$ 28,000	\$ 8,000	\$ 36,000	\$ 48,000	\$ 48,000	\$ 36,000
IBM Web Sphere ESB (8)	\$ 21,075	\$ 8,430	\$ 126,450	\$ 50,580	\$ 48,000	\$ 18,240
Web Sphere Service Registry	\$ -	\$ -	\$ -	\$ -	\$ -	\$ 9,120
Web Sphere Message Broker	\$ 71,655	\$ 28,662	\$ 214,965	\$ -	\$ -	\$ 9,120
Total Cost	\$ 92,730	\$ 37,092	\$ 341,415	\$ 50,580	\$ 48,000	\$ 36,480
Mule	\$ -	\$ 12,000	\$ -	\$ 12,000	\$ 48,000	\$ 18,240
Total Cost	\$ -	\$ 12,000	\$ -	\$ 12,000	\$ 48,000	\$ 18,240
Year 2 and 3 Cost Analysis						
Cost Analysis - Other Costs	Additional License Cost (4)	Additional Maintenance Cost (5)	Additional Server Cost (4)	Ongoing Training Costs	Reserve for Other Expense (7)	ALL COSTS 3 YEAR TOTAL
Apache Service Mix	\$ -	\$ -	\$ 16,000	\$ 9,120		
Total Cost	\$ -		\$ 16,000	\$ 9,120	\$ 2,741	\$ 94,101
AquaLogic Service Bus	\$ 77,000	\$ 29,400	\$ 16,000	\$ 31,920		
Total Cost*	\$ 77,000	\$ 29,400	\$ 16,000	\$ 31,920	\$ 48,195	\$ 1,654,680
Fiorano	\$ 56,000	\$ 16,000	\$ 16,000	\$ 18,240		
Total Cost	\$ 56,000	\$ 16,000	\$ 16,000	\$ 18,240	\$ 9,307	\$ 319,547
IBM Web Sphere ESB (8)	\$ 113,805	\$ 37,092	\$ 16,000	\$ 27,360		
Total Cost	\$ 113,805	\$ 37,092	\$ 16,000	\$ 27,360	\$ 24,017	\$ 824,571
Mule		\$ 24,000		\$ 9,120		
Total Cost	\$ -	\$ 24,000	\$ 16,000	\$ 9,120	\$ 4,181	\$ 143,541

(1) Costs supplied by Vendors

(2) Assumes initial environments for Acceptance Testing, Production, and Development (2-Linux Dual Processor Servers, 4GB Memory, 1.2TB HD) for each environment

(3) Assumes initial training for 6 FTE Analysts at quoted vendor rates

(4) Assumes an addition of one Linux Dual Processor Server for years 2 and 3 for Production and Acceptance Testing

(5) Assumes support and maintenance cost per Vendor price schedules

(6) Assumes additional training for 6 FTE Analysts per year

(7) Assumes a 3% of total cost contingency for unplanned expenses

Table 8: Three Year Cost Comparison

Products with more complexity and component offerings will generally have a correspondingly higher expense for training State technical personnel. For purposes of equity, no attempt was made to assess variations in training complexity from a time perspective with State personnel. Each of the options has some reserves for contingencies. Final cost information will vary, as final deployment configurations are established. In most cases costs will be reduced.

CONCLUSIONS

There are many good ESB alternatives available for use by the State. Most of the vendors recommend a phased evolutionary implementation approach to ensure buy-in and to provide sufficient time to establish effective governance and management practices.

To move toward the actual selection of one or more usable products, a more detailed use-case driven evaluation should be implemented as described in Appendix A. This will provide information suitable for reaching actual deployment and configuration recommendations.

As SOA and ESB projects are more clearly defined, the State should concurrently begin the establishment of the SOA governance group to ensure that appropriate governance and management controls are in place for all of the SOA aspects previously identified.

Once a decision has been made to move toward the SOA ESB implementation as a long term strategy, ongoing training and communication will need to be established for all of the State's employees and contractors that develop applications.

The long term benefits to the State, in terms of business value and greater efficiency, are significant, and are consistent with what most have identified as a key future best practice for developing, sharing, and integrating applications and services across the State enterprise.

APPENDIX 1: SCENARIO/USE CASE BASED PRODUCT COMPARISONS

The evaluation framework recommended for making detailed scenario or use-based ESB comparisons is illustrated in Figure 7. This framework consists of six major sections and is based upon the Seybold framework.³³

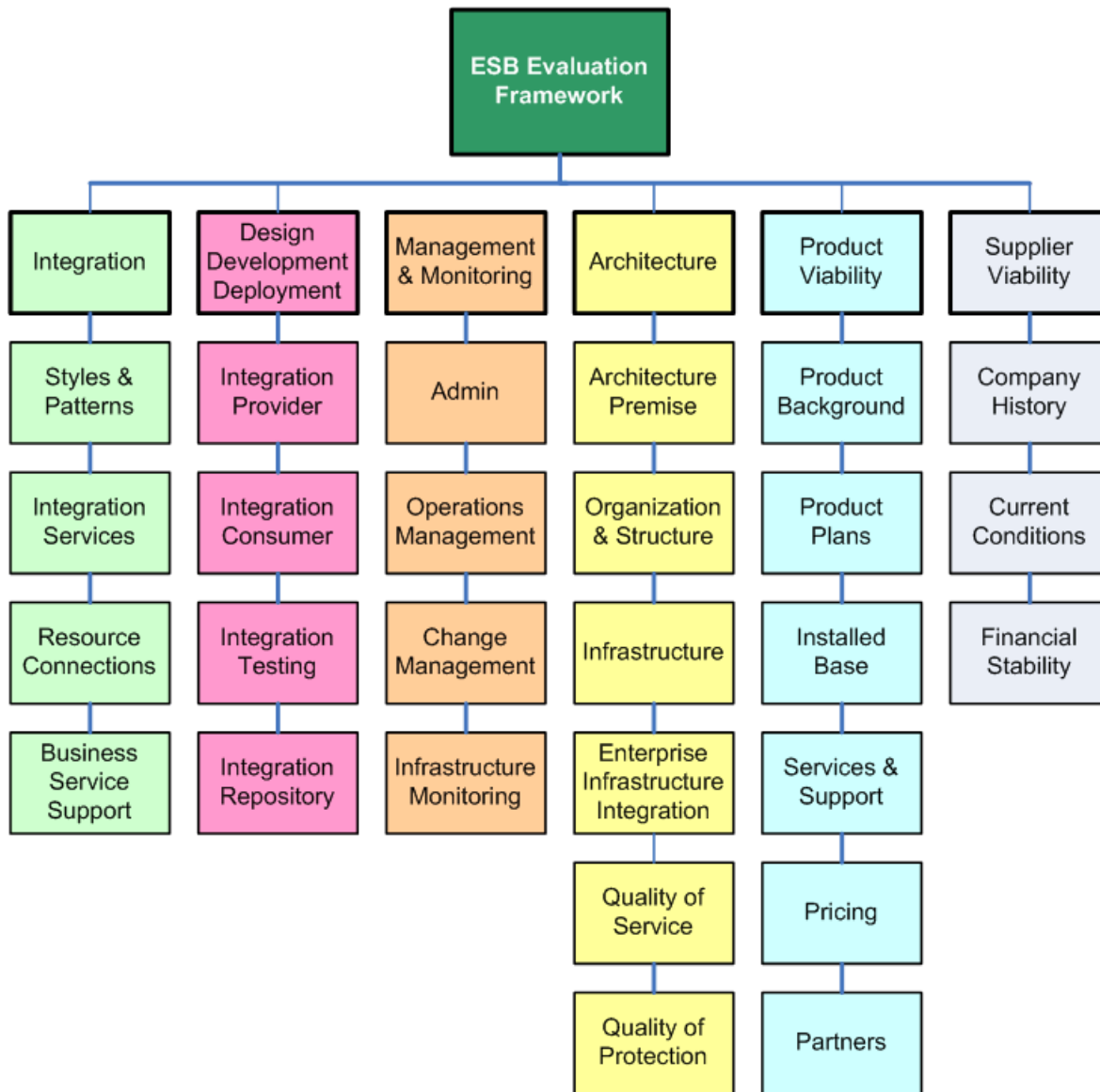


Figure 7: ESB Evaluation Framework

³³Michelson, Brenda, *Enterprise Service Bus Evaluation Framework: Criteria for Selecting an Enterprise Service Bus as an Integration Backbone*. Boston: Patricia Seybold Group, July 28, 2005.

This detailed evaluation is intended to be completed as part of a proof of concept activity with at least three selected vendor solutions. BEA AquaLogic and Mule have been selected as two of the ESB platforms for proof of concept evaluation.

Categories, definitions, and names for integration patterns have been taken from generally accepted sources, such as Hohpe and Woolf's *Enterprise Integration Patterns*³⁴ and Chappell's *Enterprise Service Bus*.³⁵ The criteria and capabilities definitions are quoted from Seybold's *Enterprise Service Bus Evaluation Matrix*.³⁶

CRITERIA AND CAPABILITIES

The modified Seybold ESB evaluation criteria illustrated in Figure 7 include evaluation with required (and suggested) capabilities in the following six areas:

- **Integration:** Integration represents the functional requirements for the enterprise service bus. Specifically, the types of scenarios that can be delivered, the types of resources that can be connected, and the building blocks (integration patterns, integration services) provided to facilitate scenario delivery.
- **Design, Development, and Deployment:** Design, development, and deployment looks at how to deliver integration scenarios. This includes tools for integration providers, integration consumers, and testers. Feeding the integration tools is a repository.
- **Management and Monitoring:** Management must occur on two levels: the integration scenario (application) level, and the ESB infrastructure (systems) level. Monitoring must occur on three levels: the integration scenario, the ESB infrastructure, and the business level.
- **Architecture:** The objectives of the architecture evaluation are to understand how the ESB works, its fit within the target environment, and its fit within the architecture. To do this, the ESB's organization must be reviewed, as well as the interoperability of its parts, deployment environment requirements, enterprise infrastructure dependencies (and conflicts), and its capabilities for quality of service and quality of protection. In addition, to provide insight into architectural direction fit, the architectural premise of the solution must be reviewed. In other words, what are the ESB creator's views on the challenges of integration and the future of services?
- **Product Viability:** Product viability criteria consider the business aspects of enterprise service buses and their suppliers.
- **Company Viability:** A company's history and current financial statistics are key markers for its future viability.

³⁴ Hohpe, Gregor, and Woolf, Bobby., *Enterprise integration patterns: designing, building, and deploying messaging solutions*, Addison-Wesley, 2004, 683 p.

³⁵ Chappell, David A., *Enterprise Service Bus*, O'Reilly, 2004, 247 p.

³⁶ Michelson, Brenda M., *Enterprise Service Bus Evaluation Matrix: A Blank Matrix to Facilitate Your Evaluation*, Patricia Seybold Group, August 11, 2005.

Integration Scenario Evaluation

Evaluation Criteria: Integration Styles		
Enterprise Service Bus Capabilities		
<ul style="list-style-type: none"> What styles of integration does this solution support? <ul style="list-style-type: none"> Information Exchange Publication/Subscription Event Processing Business/Process Execution Service Orchestration What styles of integration are implemented at the vendor's reference accounts? 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Scenarios		
Enterprise Service Bus Capabilities		
<ul style="list-style-type: none"> Describe the critical scenarios you want to resolve with the ESB. 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: State Integration Scenario/Use Case: DPS		
Enterprise Service Bus Capabilities		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: State Integration Scenario/Use Case: UMD Web Services		
Enterprise Service Bus Capabilities		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: State Integration Scenario/Use Case: UDDI/Service Directory		
Enterprise Service Bus Capabilities		
Products		
TBD	BEA AquaLogic	Mule

Evaluation Criteria: Integration Services		
Enterprise Service Bus Capabilities		
<ul style="list-style-type: none"> • What composition and choreography services are available with the ESB, or by third parties? <ul style="list-style-type: none"> • Business process support orchestration (BPEL) • Workflow • Information dissemination • Event processing: simple, complex, event stream • Scenario-level security (authentication, authorization) • Scenario-level policy • Scenario-level coordination/compensation, state • What backbone integration services are available with the ESB, or by third parties? <ul style="list-style-type: none"> • Routing (addressability, content-based routing, synchronous transport (HTTP/S), asynchronous transport (JMS, WS-RM)) • Invocation Protocol Support: How do you invoke a service on the bus, which is different from the types of resources (and their protocols) that can attach to the bus, and the native protocol used by the bus? (WS-I (SOAP, UDDI, WSDL), XML/HTTP, JMS, JCA) <ul style="list-style-type: none"> • Mediation (data transformation and translation (XSD, DTD, XSLT, XPath), protocol translation (SOAP, JMS, WS-RM, JCA, SMTP), security translation) • Transaction Control • Service/Resource Level Security (authentication, authorization) • Audit • What system-oriented integration services are available with the ESB, or by third parties? <ul style="list-style-type: none"> • Business Rule Processing • Business Vocabulary/Dialect Support (ebXML, HIPAA, SWIFT, UBL, XBRL) • Data translation, Data Transformation, Enrichment, Semantic Interpretation, Data Validation • Data Extraction, Data Load • Context-based Routing, Context Filtering • Publication and Subscription • Dispatch and Aggregation; Split and Join • Notification/Alert/broadcast (e-mail, IM, page, RSS) • Scheduling, Timing • Sequence and Re-sequence <ul style="list-style-type: none"> • Persistence, Caching • Event Interpretation, Event Correlation, Event Pattern Matching • What business-oriented integration services are available with the ESB, or by third parties? <ul style="list-style-type: none"> • Trading Partner Services, File Drop • Business Calendar • Business Directory (organization, roles, groups) • Data Cleansing • Is there support for custom integration service development? 		
Products		

TBD	BEA AquaLogic	Mule
Evaluation Criteria: Resource Connections		
Enterprise Service Bus Capabilities		
<ul style="list-style-type: none"> How are resources advertised, attached, detached? How does a resource (as a provider) receive a request? How does a resource (as a provider) fulfill a request? How does a resource (as a consumer) make a request? How does a resource (as a consumer) receive a reply? What are the native resource protocols? What resource connections are available through adapters? 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Business Service Support		
Enterprise Service Bus Capabilities		
<ul style="list-style-type: none"> Does this solution allow for the creation and deployment of business services into the integration container? 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Samples		
Enterprise Service Bus Capabilities		
<ul style="list-style-type: none"> What samples are available? Service Definitions Integration Service Examples: XSLT, BPEL, Routing. Resource Connection Examples: Services, Database, JCA, JMS. Resource Examples: Web Service, Java Service, .Net Service, FTP. Where are the samples, in the installation software, on the Web? Are they current? 		
Products		
TBD	BEA AquaLogic	Mule

Design, Development, and Deployment Evaluation

Evaluation Criteria: Integration Provider Tools

Enterprise Service Bus Capabilities

- Can you define an integration service from scratch, from an integration pattern, from existing code elements (Java, .Net, COBOL, CORBA), or from data schemas?
- Can you generate proxies and skeleton code for a service?
- Can you design and code (or configure) the integration service?
- Can you deploy the integration service?
- Can you manage versions of service, integrated with source code management tools?
- Can you retire (obsolete) an integration service?
- Can you identify and attach resources to the ESB?
- Can you detach resources from the ESB?
- Can you define policy and service-level information for integration services and integrated resources?

Products

TBD

BEA AquaLogic

Mule

Evaluation Criteria: Integration Consumer Tools

Enterprise Service Bus Capabilities

- Can you define an integration scenario from scratch, an integration pattern, or existing integration scenario?
- Can you identify and connect to resources?
- Can you identify and consume integration services?
- Can you define policy and service-level information for the scenario?
- Can you deploy the integration scenario?
- Can you manage versions of the integration scenario, integrated with source code management tools?
- Can you retire (obsolete) an integration scenario?

Products

TBD

BEA AquaLogic

Mule

Evaluation Criteria: Integration Testing

Enterprise Service Bus Capabilities

- Capability for local unit and scenario-level test and debug?
- Capability for distributed scenario testing and debugging?
- Visibility into performance attributes?
- Capability for load/stress testing, using stored scripts, simulating volume (normal, spike, low), and message payload?

Products

TBD

BEA AquaLogic

Mule

Evaluation Criteria: Integration Repository**Enterprise Service Bus Capabilities**

- Does the service metadata include name, purpose, classification, disposition, current uses, and service levels?
- Can you advertise your integration services to both integration consumers and integration providers?
- Can you advertise your integrated resources?
- Can you discover resources in the enterprise IT environment that are integration candidates?
- Can you do impact analysis, at the resource, integration service, and scenario levels?
- Can you feed your runtime environment (registry and any enterprise policy and service-level operations)?
- Can you interface with all IT metadata management environments (business terms, taxonomies, schemas, business rules, processes, event definitions, event routings, integration routings, etc.)?
- Can you leverage the repository information for capacity planning and service performance analysis?

Products

TBD	BEA AquaLogic	Mule

Management and Monitoring Evaluation**Evaluation Criteria: Administration****Enterprise Service Bus Capabilities**

- Is there a unified administration console?
- Is there remote access to the administration console? Does it require a client-side implementation?
- Are there prepackaged scripts or functions for startup, shutdown, restart, log rolling, cache management, backup, and recovery?
- How difficult is the installation?
- How frequently do patches come out?
- How frequently do maintenance releases come out? Are they all-inclusive of patches? Do they align with infrastructure dependencies (application server, messaging infrastructure, operating systems)?
- Can user rights (administration, monitoring, deployment, execution) be granted by role? Does it tie to an existing LDAP scheme?

Products

TBD	BEA AquaLogic	Mule

Evaluation Criteria: Operations Management**Enterprise Service Bus Capabilities**

- Can you easily start, stop, and restart specific ESB components?
- Can you easily start, stop, and restart integration services? Integration scenarios? Resources? What happens to work in flight?
- Can you easily start, stop, and restart an executing instance of an integration scenario? What happens to the work? Does it roll back?
- How do you know there is a problem? Are infrastructure problems fed to your enterprise systems management tools? Are integration scenario problems fed to your enterprise application management tools?
- How difficult is it to isolate the problem cause? What tools and trails are provided?
- Can you find an answer quickly? What information does the vendor provide (support documentation, knowledgebase, FAQ, technical support desk)? What information does the user community provide (Google Groups)?

Products

TBD

BEA AquaLogic

Mule

Evaluation Criteria: Change Management**Enterprise Service Bus Capabilities**

- Can you do hot deployment?
- Can you have multiple versions of an integration service, integrated resource, or integration scenario in production?
- Can you label an integration service, integrated resource, or integration scenario for lifecycle management (development, testing, QA, production)?
- Can you back out a version? Easily revert to prior version?

Products

TBD

BEA AquaLogic

Mule

Evaluation Criteria: Scenario Monitoring**Enterprise Service Bus Capabilities**

- Can you instrument at the scenario, integration service, and resource connection level for errors, audit, and usage?
- Can you employ real-time monitors to notify you for specific conditions? Does that notification tie into enterprise notification/collaboration systems?
- Can you easily interpret the instrumentation output data schema?
- Is the instrumentation data accessible for use by an analytic tool? If not, can you (easily) extract and load it to an analytical data store? Can you leverage the ESB to move this data?
- Does the ESB provide analytic tools?

Products

TBD

BEA AquaLogic

Mule

Evaluation Criteria: Infrastructure Monitoring**Enterprise Service Bus Capabilities**

- Can you discern what specific process is having a problem?
- Is the instrumentation output in a standardized management protocol, such as SNMP, CIM, JMX, and WSDM?
- Can you set thresholds for utilization and performance, and then specify actions to occur automatically when a threshold is hit?

Products

TBD	BEA AquaLogic	Mule

Architecture Evaluation**Evaluation Criteria: Architectural Premise****Enterprise Service Bus Capabilities**

- What is the provider's architectural premise?
 - Do they view a future of heterogeneous applications, information, and services?
 - Do they view a future of homogenous assets, built around the Web Services specifications?
 - Do they have an RPC (verb) view of services? Services provide action, via an operation.
 - Do they have a REST (noun) view of services? Services deliver and manage documents.
 - Do they emphasize technical mediation challenges over business and data mediation challenges?
 - Do they believe the ESB is smart? Or do they believe the smarts are at the endpoints?
- Do they account for a human in integration? Do humans perform actions in an integration scenario? Do humans initiate an integration scenario? Is the human (user interface) a type of integration service?

Products

TBD	BEA AquaLogic	Mule

Evaluation Criteria: Organization**Enterprise Service Bus Capabilities**

- What are the components of the ESB?
- Do you have choices?
- How do the components interoperate?

Products

TBD	BEA AquaLogic	Mule

Evaluation Criteria: Structure**Enterprise Service Bus Capabilities**

- How is the ESB built?
 - What are the parts—router/bus, services, connections, container, and directory?
 - Does it use standard technology in its implementation (J2EE, XML, WS-RM)?
- Does the ESB conform to the Java Business Integration (JBI) specification? Are there plans to?
- What is the native bus message format and protocol? How does the bus invoke services? How do services communicate back to the bus? Do services ever communicate directly (without the bus)?
- How do resources attach to the bus? How do the adapters work? How many translations are involved?
- What is the standard interaction pattern?
- What are the format, storage mechanism, and interpreter of the integration scenario? BPEL? XML itinerary?
- What does an integration scenario execution look like? How is it coordinated? Managed?
- Where is execution state held?
- How smart is the bus? How smart are the endpoints?
- How are integration services implemented?
- How are (if applicable) business services implemented? Invoked?

Products

TBD	BEA AquaLogic	Mule

Evaluation Criteria: Infrastructure**Enterprise Service Bus Capabilities**

- What are the deployment environment requirements for the integration tooling?
- What are the deployment environment requirements for the management console?
- What are the deployment environment requirements for the ESB?

Products

TBD	BEA AquaLogic	Mule

Evaluation Criteria: Enterprise Infrastructure Integration

Enterprise Service Bus Capabilities

- Does the ESB conflict with, or leverage, any of your following enterprise assets:
 - Enterprise Messaging
 - Directory (LDAP)
 - Backup and Recovery
 - Security Models
 - Systems and Application Management Tools (JMX, SNMP, CIM)
 - Development Tools (Eclipse)
 - Testing Tools
 - Source Code Management
 - Repository
- Registry

Products

TBD	BEA AquaLogic	Mule

Evaluation Criteria: Quality of Service

Enterprise Service Bus Capabilities

- How does the ESB provide for basic QOS?
 - Transaction Control and Compensation
 - Fault Avoidance and Tolerance (exception handling (catch, notify, compensate), failover at service and service container (clustering))
 - Performance and Scale (load balancing, multithreading, caching, XML acceleration, flexible [small footprint] deployment, service, and scenario prioritization)
- How can the ESB be configured (settings, topology options) to increase QOS?
- Does the ESB provide the QOS directly, or does it leverage functionality in the underlying enterprise infrastructure (J2EE application server, messaging infrastructure, network, hardware, etc.)?

Products

TBD	BEA AquaLogic	Mule

Product Viability

Evaluation Criteria: Quality of Protection		
Description		
<ul style="list-style-type: none"> Is the ESB environment secure? <ul style="list-style-type: none"> Do you need rights to deploy to the ESB? Do you need rights to administer the ESB? Do you need rights to start/stop an integration scenario instance? Can you see cached transaction information, or any other internal persistence mechanisms? What encryption/decryption is supported? Can you see message payloads? What encryption/decryption is supported? Does the ESB comply to (and enforce) the security model (authentication, access control) of integrated resources and external endpoints? Does the architecture allow for a DMZ deployment? What security models are used? How are policies managed? Are there mechanisms or controls for compliance (SOX, HIPAA)? 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Product Background: Release History, Next Release, and Timeframe		
Description		
<ul style="list-style-type: none"> Current Version, Month, and Year Previous Version, Month, and Year Previous Version, Month, and Year (etc.) First Release Version, Month and Year Version, Quarter, and Year Currently Planned 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Product Plans		
Description		
<ul style="list-style-type: none"> Frequency of Major and Minor Releases Next Planned Release Planned Enhancements for Next Release 		
Products		
TBD	BEA AquaLogic	Mule

Evaluation Criteria: Installed Base		
Description		
<ul style="list-style-type: none"> Number of Customers for this Product Number of Paying Customers (if substantially different) Examples of Customers OEM Relationships 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Target Market		
Description		
<ul style="list-style-type: none"> Industries 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Services and Support		
Description		
<ul style="list-style-type: none"> Training Documentation Product Support User Communities Proof-of-concept Trials 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Pricing		
Description		
<ul style="list-style-type: none"> Average Pricing or Base Pricing Pricing for Options, Modules, or Complementary Products Discussed in the Report 		
Products		
TBD	BEA AquaLogic	Mule

Evaluation Criteria: Competition		
Description		
<ul style="list-style-type: none"> Alternatives (How does this solution compare with the alternatives?) 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Partners		
Description		
<ul style="list-style-type: none"> Third-Party Offerings (integration services, adapters, test tools, analytics) Support Training Consulting (implementers) 		
Products		
TBD	BEA AquaLogic	Mule

Company Viability

Evaluation Criteria: Company History		
Description		
<ul style="list-style-type: none"> Date Founded Founders and Investors (if the company is young) Community Leaders, backers, and Funding (if open source product) Location of Headquarters Number of Employees Number of Customers 		
Products		
TBD	BEA AquaLogic	Mule
Evaluation Criteria: Current Conditions		
Description		
<ul style="list-style-type: none"> Exceptional Market Conditions Investor or Regulatory Actions Recent or Anticipated Structural Changes 		
Products		
TBD	BEA AquaLogic	Mule

Evaluation Criteria: Financial Stability		
Description		
<ul style="list-style-type: none">Last four quarters of revenue and income, as available.		
Products		
TBD	BEA AquaLogic	Mule

DEFINITIONS

.NET

A comprehensive software development platform from Microsoft that was introduced in 2000 as the company's next generation programming environment.

Adapter API

An adapter used as a Web service with an application's API.

API

Application Programming Interface is a language and message format used by an application program to communicate with the operating system or some other control program such as a database management system (DBMS) or communications protocol.

ACEGI

ACEGI Security provides applications with comprehensive authentication, authorization, instance-based access control, channel security, and human user detection capabilities.

ActiveMQ

ActiveMQ is an open source (Apache 2.0 licensed) JMS message broker.

Axis

Apache Axis is a SOAP stack that not only supports SOAP 1.1 and SOAP 1.2, but it also has integrated support for the REST style of Web services.

BPEL

Business Process Execution Language is an XML-based language developed by IBM, BEA Systems, Microsoft and others for defining Web services business processes.

C/C++

Refers to both object oriented programming language environments.

CIM

Common Information Model is a model for describing management information from the Distributed Management Taskforce (DMTF).

DMZ

DeMilitarized Zone is a middle ground between an organization's trusted internal network and an untrusted, external network such as the Internet.

DTD

Document Type Definition is a language that describes the contents of an SGML or XML document.

ebXML

Electronic Business using XML is a framework for developing a business transaction vocabulary based on XML.

Eclipse

An open source Java-based platform for integrating software tools for application development.

EDA

Enterprise Data Access provides a uniform way to access data throughout the enterprise. It implies the ability to treat multiple, distributed databases as a single logical entity.

EJB

Enterprise Java Bean is a software component in Sun's J2EE platform, which provides a pure Java environment for developing and running distributed applications.

ESB

Enterprise Services Bus is a message and/or integration broker that supports Web services.

HIPAA

Health Information Portability and Accountability Act that governs the privacy and security of health information records and transactions.

J2C

The connector architecture specification (JCA Specification) is a standard architecture for integrating Java applications with existing enterprise information systems.

J2EE

Java 2 Platform, Enterprise Edition is a platform from Sun for building distributed enterprise applications.

JAAS

Java Authentication and Authorization Service is an API that enables Java applications to access authentication and access control services without being tied to those services.

Java

An object-oriented programming language that is platform independent.

JB1

Java Business Integration (JB1) specification.

jBPM: Java Business Process Management (jBPM) is a flexible, extensible workflow management systems and is a set of J2SE components that can also be deployed as a clustered J2EE application.

JCA

Java Connector Architecture

JDBC

Java DataBase Connectivity is a programming interface that lets Java applications access a database via the SQL language.

JMS

Java Messaging Service is a programming interface (API) from Sun for connecting Java programs to messaging middleware

JMX

Java Management Extensions or JMX is a Java technology that supplies tools for managing and monitoring applications, system objects, devices (e.g. printers) and service oriented networks.

JNDI

Java Naming and Directory Interface is a programming interface from Sun for connecting Java programs to naming and directory services such as DNS, LDAP and NDS.

JOTM

Java Open transaction Manager is a transaction manager written in Java and released under an Open Source license, such as LGPL, GNU Lesser General Public License.

JSR 208

Java Specification Request 208 defines the core of a service oriented integration bus and component architecture for SOA.

JTA

Java Transaction API is a programming interface from Sun for connecting Java programs to transaction monitors, such as IBM's CICS and BEA's Tuxedo. JTA is part of Sun's J2EE platform.

LDAP

Lightweight Directory Access Protocol is a protocol used to access a directory listing.

NASCIO

National Association of Chief Information Officers

ODBC

Open DataBase Connectivity is a database programming interface from Microsoft that provides a common language for Windows applications to access databases on a network.

PGP

Pretty Good Privacy is a data encryption program from PGP Corporation that is widely used for encrypting e-mail messages and securing files.

PicoContainer

PicoContainer is a lightweight and highly embeddable container for components that honor Dependency Injection.

Plexus

Plexus provides a full software stack for creating and executing software projects. Founded on the Plexus container, applications can utilize component-oriented programming to build modular, reusable components that can easily be assembled and reused.

POJO

Plain Old Java Object is an object that was created as a regular Java class and is not a JavaBean or EJB.

REST

Representational State Transfer Web services are resource oriented Web services. Resource-oriented services focus on distinct data objects upon which a handful of basic, standard operations can be performed.

RPC

Remote Procedure Call is a programming interface that allows one program to use the services of another program in a remote machine.

SNMP

Simple Network Management Protocol is a widely used network monitoring and control protocol.

SOA

Service-Oriented Architecture was formerly called a "distributed objects" architecture, the SOA term was coined as Web services were evolving.

SOAP

Simple Object Access Protocol is a message-based protocol based on XML for accessing services on the Web.

SOX

Schema for Object-oriented XML is an XML schema based on DTD, but adds data typing and reuse mechanisms.

Spring

An application development framework.

SWIFT

An industry cooperative that provides a standard format for transmitting payments, stock transactions, letters of credit and other financial messages to more than member banks, broker-dealers and investment organizations around the world.

TCO

Total Cost of Ownership is the cost of using a computer or information system.

UBL

Universal Business Language is a format for exchanging data from one XML business language to another.

UDDI

Universal Description Discovery Integration is designed to enable software to automatically discover and integrate with services on the Web.

Web Services

Web-based applications that dynamically interact with other Web applications using open standards that include XML, UDDI and SOAP.

WSDL

Web Services Description Language is an XML-based language for defining Web services.

WSDM

Web Services Distributed Management

WS-I

Web Services Interoperability Organization

WS-REL

Web Services-Reliability defines an open interoperable wire protocol for reliable messaging based on the SOAP protocol.

WSRM

Web Services Reliable Messaging specifies a generic and open model for ensuring reliable message delivery for Web services.

WSRP

Web Services for Remote Portlets are dynamic plug-ins for portal pages. WSRP defines how to plug remote web services into the pages of online portals and other user-facing applications.

XBRL

EXtensible Business Reporting Language is a specification for publishing financial information in the XML format.

XML

EXtensible Markup Language is an open standard for describing data from the W3C.

XPath

XML PATH is a sublanguage in an XSL style sheet that is used to identify XML elements for processing.

XSD

XML Schema Definition is the informal name for the XML schema from the W3C.

XSLT

eXtensible Stylesheet Language Transformation is software that converts an XML document into another format such as HTML, PDF or text.

REFERENCES

- Apache ServiceMix Web Site. <http://incubator.apache.org/servicemix/main/home.html>
- BEA AquaLogic Product Family: A suite of Service Infrastructure products for successfully deploying Service-Oriented Architecture in heterogeneous environments, BEA Systems, June 2005.
- BEA Overview, PowerPoint Presentation prepared by Tony Galindo, October 10, 2006.
- Chappell, David A., *Enterprise Service Bus*, O'Reilly, 2004.
- Cobb, Edward, *BEA, Standards, and Open Source*, BEAWorld2006 Presentation, September 2006.
- Commonwealth of Kentucky, *NASCIO Recognition Award Application: Enterprise Service Bus*, June 2006 (Unpublished).
- Correia, Joanne M, et al., *Forecast: AIM and Portal Software, Worldwide, 2005-2010 (Executive Summary)*, Gartner Research G00138499, March 22, 2006.
- Enterprise Service Bus and SOA Middleware*, Boston: Aberdeen Group, June 2006.
- Enterprise Service Bus (ESB) Learning Guide*. SearchWebServices.com, May 2005
http://searchwebservices.techtarget.com/general/1,295582,sid26_gci1085711,00.html
- "The ESB in the Land of SOA", *Information Technology Research: Enterprise Strategies*, Boston:Aberdeen Group, December 7, 2005.
- "Enterprise Service Bus: The Foundation of Successful Service-Oriented Architecture", *Information Technology Research Brief*, Boston: Aberdeen Group, April 27, 2006.
- Fiorano Enterprise Service Bus Versus the Competition, Los Gatos, CA: Fiorano Software Inc., 2003.
- Herr, Michael, *Enterprise Service Bus: The Next Generation*, SOP Group. San Francisco, September 19, 2005.
- Hohpe, Gregor, and Woolf, Bobby, *Enterprise integration patterns: designing, building, and deploying messaging solutions*, Addison-Wesley, 2004.
- Howard, Philip, *SOA and Information Services*, Bloor Research, March 2006.
- Hutchinson, Beth, et al., *Increasing IT Flexibility with IBM Web Sphere ESB software*, IBM, December 2005.
- Ibarra, Fausto. *The Enterprise Service Bus: Building Enterprise SOA*. BEA, December
http://dev2dev.bea.com/pub/a/2004/12/soa_ibarra.html
- The Legacy Application Modernization Benchmark Report: Transforming Mainframe, AS/400, and Unix Applications in an SOA World*, Boston: Aberdeen Group, September 2006.

- Majumdar, Biljoy, et al, *ESB-A Bandwagon worth jumping on*, Infosys Technologies Limited, May 2006.
- Michelson, Brenda, *Enterprise Service Bus Evaluation Framework: Criteria for Selecting an Enterprise Service Bus as an Integration Backbone*. Boston: Patricia Seybold Group, July 28, 2005.
- _____, *Event Driven Architecture Overview*. Boston: Patricia Seybold Group, February 2, 2006.
- _____, "SOA and Integration Infrastructure, Understand the Provider's Intent (architectural premise)" *Elemental Links*, August 2, 2005.
- Mule Architecture Guide*. <http://mule.codehaus.org/wiki/display/MULE/Architecture+Guide>
- Mule User Guide*. <http://mule.codehaus.org/wiki/display/MULE/User+Guide>
- Mule Web Site*. <http://mule.mulesource.org/wiki/disply/MULE/Overview>
- Plummer, Daryl C., *Six Missteps That Can Result in SOA Strategy Failure*, Gartner Research G00126721, June 8, 2005.
- _____, *Software Architectures Will Evolve From SOA and Events to Service Virtualization*, Gartner Research G00162246, March 9, 2005.
- Real World Labs Report Care: ESB Suites at <http://i.cmpnet.com/nc/1705/graphics/1705f2report.gif> 2006.
- Robinson, Rick. *Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part 1: The role of the Enterprise Service Bus*. IBM, June 2004. <http://www-128.ibm.com/developerworks/xml/library/ws-esbscen/>
- _____. *Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part 2: ESB scenarios and issues driving the architecture*. IBM, June 2004. <http://www128.ibm.com/developerworks/webservices/library/ws-esbscen2.html>
- _____. *Understand Enterprise Service Bus scenarios and solutions in Service-Oriented Architecture, Part 3: Solutions for ESB scenarios*. IBM, June 2004. <http://www-128.ibm.com/developerworks/webservices/library/ws-esbscen3/>
- The Role of an Enterprise Service Bus in an SOA*, Tibco: Palo Alto California. July 2006, 10 p.
- Sonic ESB: An Architecture and Lifecycle Definition*, Sonic Software Corporation, 2005.
- State of the Portal Market 2006: Portals and the New Wisdom of the Enterprise*. BEA Systems, May 2006.
- Vollmer, Ken, and Gilpin, Mike, The Forrester WaveTM: *Enterprise Service Bus*, Q2 2006, Forrester Research.
- Washington D.C., *NASCIO Recognition Award Application: Enterprise Architecture District Enterprise Integration Stack*, June 2006 (Unpublished).
- Welsh, Michael; David Cutler and Eric Brewer. *SEDA: An Architecture for Well-Conditioned, Scalable Internet Services*. Computer Science Division, University of California, Berkely. 2001. 14p.

Wikipedia. *Enterprise Service Bus*. http://en.wikipedia.org/wiki/Enterprise_service_bus

Woolf, Barry, *Introduction to SOA Governance*, IBM, June 13, 2006.

Wright, Brad, *Get on the Information Bus: Driving SOA Success*, BEAWorld2006 Presentation, September 2006.